

[Icc-avr] keypad debounce code

Alan Smewing icc-avr@imagecraft.com

Thu, 4 Mar 2004 09:14:19 -0000

- Previous message: [\[Icc-avr\] keypad debounce code](#)
- Next message: [\[Icc-avr\] GPS and mega128](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

For debouncing I use a clever little thingy called a 'vertical stack'. Instead of wasting my external interrupts I have a general purpose timer interrupt the reads the buttons and debounces them. The beauty of the vertical stack is that there are no funny loops or conditionals involved just logic. Sounds complex but have a look at the code here and you'll see that it's quite simple (the comments make it longer):

```
/******  
  
// A global which tells us if the button has been pressed and is being held  
unsigned char DebouncedState = 0;  
// A global which tells if the button has been pressed and then released  
unsigned char ButtonReleased = 0;  
void GeneralTimeOut(void)  
{  
    static unsigned char OldDebouncedState = 0; //Holds the previous value of  
the buttons.  
    static unsigned char CountA = 0; // Used in the debouncing routine  
    static unsigned char CountB = 0; // Used in the debouncing routine  
    unsigned char CurrentSample = 0;  
  
    // We enter this interrupt every 20ms!  
  
    // ***** DEBOUNCE BUTTONS *****  
    // This code does the debouncing routine using a vertical stack  
    // it's all based around logical functions.  
  
    // The Debouncing Bit  
    // Sample the buttons (from Port B)  
    CurrentSample = ~PORTA;    // Note that I invert the input, it depends on  
                                // the pull-ups on your circuit.  
  
    // Decrement the vertical counter and reset if button state changes  
    // The counter starts at 0 then counts 3, 2, 1 and back to zero at which  
point  
    // it sets the appropriate debounced state bit.  
  
    // These two lines implement a 2-bit vertical counter  
    CountA ^= CountB;  
    CountB = ~CountB;  
  
    // These two lines reset the counter if the state changes  
    CountB &= (CurrentSample ^ DebouncedState);  
    CountA &= (CurrentSample ^ DebouncedState);
```

```
// These two lines change the values of debounced state to current sample
but
// only if the vertical counter has rolled over.
DebouncedState &= (CountA | CountB);
DebouncedState |= (CurrentSample & ~(CountB | CountA));

// The statement below reads thus:
// If (OldDebouncedState = 1 AND DebouncedState = 0) OR ButtonReleased = 1
// THEN ButtonReleased = 1
// The purpose of this statement is to differentiate between pressing a
button
// and holding a button down. ButtonReleased is set when the a button has
// been pressed and released. ButtonReleased should be cleared before use
in
// the main code
ButtonReleased |= (OldDebouncedState & ~DebouncedState);

// This updates the OldDebouncedState register
OldDebouncedState = DebouncedState;
}
```

----- Original Message -----

From: Steven Williams
 To: icc-avr@imagecraft.com
 Sent: Wednesday, March 03, 2004 11:22 PM
 Subject: RE: [Icc-avr] keypad debounce code

I have used a diode array to common all key press inputs to one interrupt for interrupt driven kbd. You could also use a port with int on change of state. This allows kbd scan times to be kept to a minimum, more time for other stuff.

Best regards,
 Steve Williams
 Manager Hardware Development
 Mandalay Technologies Pty Ltd.
 E swilliams@mandalaytech.com
 P +61 (0)7 3010 7900
 F +61 (0)7 3010 7999
www.mandalaytechnologies.com.au

-----Original Message-----

From: BobGardner@aol.com [mailto:BobGardner@aol.com]
 Sent: Wednesday, March 03, 2004 10:38 PM
 To: icc-avr@imagecraft.com
 Subject: Re: [Icc-avr] keypad debounce code

In a message dated 3/3/04 4:06:11 AM Eastern Standard Time,
John.Baraclough@logitech.uk.com writes:
 > does anyone have 3x4 keypad debounce code? i do really need it.

=====

Do you have an 'anykey pressed' function? I use this like 'kbhit()' as a

quick check... turn on all columns and read the rows... if any row is pulled down (row inputs != 0xff) then a key is pressed. This is a nice quick routine.
 Right after I call 'getkey()' which turns on the columns one at a time and reads

the rows to find the actual key. Also, we know the key bounce is settled within several ms, depending on the stiffness of the spring, so I think an algo would be... check for anykey, del 3ms, check for anykey, if still pressed call getkey.

Icc-avr mailing list
Icc-avr@imagecraft.com
<http://www.dragonsgate.net/mailman/listinfo/icc-avr>

Icc-avr mailing list
Icc-avr@imagecraft.com
<http://www.dragonsgate.net/mailman/listinfo/icc-avr>

-
- Previous message: [\[Icc-avr\] keypad debounce code](#)
 - Next message: [\[Icc-avr\] GPS and mega128](#)
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)